

**AMENDMENT TO THE SPECIFICATION**

Please amend the specification by marked up replacement paragraph(s) as follows.

Please replace paragraph [07] on page 2 with the following:

-- The present invention addresses the above stated needs by providing an early payment discount mechanism for an e-billing system that provides an incentive for customer's early payment of current invoice ~~current~~ charges. --

Please replace paragraph <sup>14</sup>[19] on page <sup>4</sup>4 with the following: 4P 7-16-10

-- Figures ~~3(a) and 3(b)~~ 3(a)-3(c) are diagrams of an example electronic bill statement and presentment system 200 employed by a third-party for generating on-line invoices having the EPD incentive, according to an embodiment of the present invention; --

Please replace paragraph <sup>19</sup>[24] on page <sup>5</sup>7 with the following: 4P 7-16-10

-- As shown in Figure 1, there is depicted a physical hardware architecture of an exemplary e-billing system 100 (e.g., the MCTWorldcom<sup>TM</sup> Interact system). As shown in Figure 1, there is provided a billing system represented as cloud 110. Relating specifically to e-Billing, a link via a double firewall 125a is provided to interface the billing system 100 with a third-party vendor electronic bill statement and presentment system 200 that functions to generate and present on-line billing invoices for customers of the telecommunications enterprise over the Internet via double firewall 125b. In operation, a file including a customers billing/invoice data is provided by the Billing system (e.g., accounts receivable department) to the third-party system 200 where web-pages including corresponding versions of the customer invoices are generated

and posted on-line. Customers initially do not have access to the web based invoices. Instead, the on-line invoices are available to the enterprise audit department which checks the invoices for format compliance and billing total amount accuracy. As soon as the on-line version of the invoices are approved by the audit department, they are flagged as available on-line and the third party system 200 notifies the customer and posts the invoices on-line for the customer access. Via public Internet 420 120b access through the MCI™ "Interact" web server portal to the suite of web-based applications including the e-billing application (not shown), the customer's may pay their invoices. --

Please replace paragraph <sup>30</sup>[25] on pages <sup>6-7</sup>~~29~~ with the following: 4p 7-16-10

-- Figures 2, 3(a) - 3(c) particularly depict a data flow summary 300 and the invoice generation process involving the billing enterprise and the third-party electronic bill statement and presentment system for generating invoices and applying EPD discount according to the invention. As shown in Figure 2, a first step 303 depicts generally, the step of providing the customer invoice data to the third-party electronic billing and presentment system 200 (Figure 1). This information may be provided in flat format files to the third-party system and may comprise softcopy images of the customers paper invoices. Preferably, a process is performed whereby data is extracted from the enterprise legacy billing system 110 and transported to a common data store maintained by the third party system 200. As depicted in Figure 3(a), there are two methods in which a billing system 110 may provide bill data to the third party e-billing payment and presentment system 200: by either providing a filtered extract 125 including only those bills destined for electronic delivery, or, they can provide a non-filtered extract 150 that includes all bills. In the latter scenario, the third-party system is required to match bill data against an enrollment store 190 as depicted in Figure 3(a). There are several formats that the billing system

may provide to the third-party system 200 including, but not limited to: AFP (IBM's<sup>TM</sup> Advanced format printing), Line Data, ~~Xerox~~ XEROX<sup>TM</sup> Metacode, and electronic data interchange (EDI) electronic file format. Once the extract has been completed by the billing system 110, the data is then routed to the third-party system 200 for processing. Implementing known techniques, a dedicated communication circuit may be used to route this data to the third party system for high volume billers, or a Value Added Network (VAN) 175 may be implemented as well for receiving the data and routing data to the billing database. Once the data has been sent to the VAN, the third-party system 200 sweeps the assigned VAN mailbox on a regular basis to obtain bill data and load it to an interim data store 180 on an electronic bill presentment and payment ("EBPP") server 195, as shown in Figure 3(c). As further shown in Figure 2, at step 309, business rule criteria for applying EPD to the invoice billing data for customers is transmitted to the third-party system for application to the invoices. As will be explained in greater detail herein, this billing data includes the discount period(s) and one or more levels of discount percentages for the billing period. Then, as shown in Figure 2, there is the step performed by the third-party system of processing the extracted customer data and generating invoices for on-line presentation. This step is depicted in greater detail in Figures 3(b) and 3(c). --

Please replace paragraph <sup>21</sup>[26] on pages <sup>7-8</sup>~~9-10~~ with the following: ~~4P~~ <sup>7-16-10</sup>

-- Referring to Figure 3(b), once customer invoice data has been loaded to the interim data store 180 on the EBPP server ~~180~~ <sup>195</sup>, it is parsed for loading into a database, e.g., ~~Oracle~~ ORACLE<sup>TM</sup> database 210. This process involves several steps, including known processes such as facilitated by ~~BlueGill<sup>TM</sup>~~ BLUEGILL<sup>TM</sup> software available from BlueGill Technologies, Inc, the content and disclosure of which is incorporated by reference herein. First, the raw input file is moved to a working directory that is monitored by a continuously running splitter process 182

in a server. Once the file arrives in this directory, the splitter 182 automatically begins the process of splitting the file into distinct statements, generating a single working file for each. Also during this process, the splitter parses each statement for key information such as the account number, invoice amount. The splitter process only generates a working file 205 for those statements whose account number matches with an account in the enrollment data store 190. This step assumes that a statement definition file 192a and an extract definition file 192b have been previously created and loaded to the server. These files 192a,b define how to locate key elements within the statement, and which elements will be extracted and loaded as a discreet bill on the server. Second, the working files 205 are input to a second working directory that is monitored by an extract process 184 continuously running on the server. The extract process begins processing statements as soon as the working files 205 arrive. The extract process parses through each work file (statement) based on the location of information defined in the extract definition, and extracts key elements that were defined in the statement definition. This information is then compressed into a Binary large object (BLOB) (except for key fields such as account and statement date) and loaded to the ~~Oracle~~ ORACLE™ database 210 as a distinct statement record to be later accessed via a web-based server interface. Once all working files 205 have been processed, the process repeats with the next extract from the billing enterprise. --

Please replace paragraph <sup>22</sup>[27] on page <sup>8</sup>10 with the following: 4p 7-16-10

-- The actual generation of the final on-line (HTML) invoice is now described with respect to Figure 3(c). As illustrated in Figure 3(c), the third-party EBPP server 195 hosts the ~~Oracle~~ ORACLE™ database 210 that warehouses all customer invoice data in a format accessible by the web-server interface. The stored bills include all necessary information to generate a complete invoice for the customer including bill summary, bill detail, and all data

elements originally defined in the statement definition. A method is additionally invoked to apply business rules for determining whether the customer may receive an EPD discount, and if determined that a EPD discount is to be applied, generated additional data including the discount amount, and the discount availability period. The primary output mechanism is a web-server interface which services requests from various processes and acts as the sentry between requesting processes, e.g., an HTML server 250 and the ~~Oracle®~~ ORACLE™ database 210. Compressed customer invoice data in the database is instantiated only on request via the interface which serves requests from a web application, or alternately from scheduled processes from a Consolidated Service Provider (CSP) that facilitates the transport of information. --

Please replace paragraph <sup>32</sup>[38] on page <sup>12</sup>13 with the following: 4p 7-16-10

-- Alternately, as shown in Figure 4(a), from a menu bar 411 providing other user e-billing system options, the user may select a "pay invoice" button 412 (not shown) for initiating payment of the invoice. --

Please replace paragraph <sup>33</sup>[39] on page <sup>12</sup>14 with the following: 4p 7-16-10

-- Upon selection of either the "pay now" button 415 or, the "pay invoice" button 412, an EPD home page 425 is displayed at the user browser as illustrated in Figure 4(b). In Figure 4(b), the EPD home page 425 includes an HTML frame 430 presenting the customer with a banner 432 explaining the EPD discount option for the customer. The frame is further populated with information 435 for enabling immediate EFT payment of the invoice by customer selection of the Pay button 440, or, make a later scheduled payment by selection of Schedule button 445. The EFT payment information is automatically populated with the bank account information, invoice date, number, total current charges, and the calculated EPD amount 437 based on the applied

percentage, and, the total amount due 439 when the discount is applied. Provided in the frame 430 is a banner 447 indicating what percentage is being applied for the EPD, and which is populated with the re-calculated date 449 indicating the time period within which the EPD is effective. It is understood that this date is calculated based on the on-line invoice posting date. Furthermore, according to the invention, it is understood that the web application executing on the BG EBPP server (Figure 3(c)) pre-populates the calculated EPD amount 437, the total amount due 439 and, the re-calculated date 449 depending upon the point in time when the customer has accessed the on-line invoice. For instance, if the user has accessed the on-line invoice 11 days after the posted invoice date, then that customer would not be eligible for the first tier of discount, e.g., 2% within ten (10) days of invoice date, and accordingly the EPD home screen 425 will not display the first tier EPD criteria. However, if the customer is within a second tier EPD criteria, the EPD home screen will be automatically populated with the second tier discount criteria, e.g., 1% within 20 days of invoice date, for example. This principle applies for each EPD tier when multiple discount periods are offered with the percentage of discount varying by the date payment is made. --

Please replace paragraph <sup>36</sup>[42] on pages <sup>14</sup>15-16 with the following: ~~AP~~ 7-16-10

-- In Figure 4(e) the scheduled EDP confirmation screen 490. Preferably, the web application executing on the BG EBPP server (Figure 3(c)) makes a determination as to whether the customer entered scheduled payment date is within the applicable EPD discount time window. If the scheduled payment date is outside the window, a banner 491 is displayed on HTML frame 492 reminding the user that the scheduled payment date was greater than the EPD discount time window, and that, the customer may reschedule the payment for prior to the